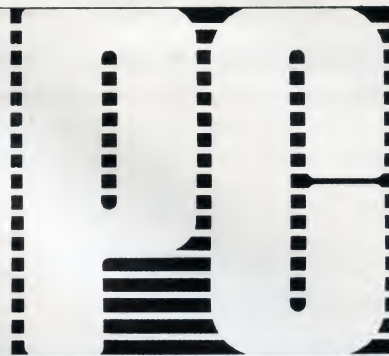


LANGUAGES/ROBERT S. LEE

A long-time mainframe APLer evalutes a new implementation by the nation's leading APL supplier.



**The Independent Guide to
IBM Personal Computers**

PC APL ication

APL*PLUS/PC
STSC, Inc.
2115 East Jefferson Street
Rockville, MD 20852
(301) 984-5000
List price: \$595

I had used APL almost daily since the late sixties while working at IBM. When I recently bought an IBM PC to use at home, I was delighted with many of its features. The crisp monochrome monitor, for example, was a vast improvement over the clattering 2741 terminal that had dominated my Armonk office. But I was very disappointed that the PC did not support

Figure 1: Informal comparison of APL*PLUS/PC and BASIC in counting and calculating cumulative sums.

N	LOOPCOUNT	RHO COUNT	SUMTO
1,000	18.5	0.1	0.8
3,000	55.6	0.2	2.3
5,000	92.7	0.3	3.7

APL IS NOT
*simply an expensive
calculator.*

APL.

Over the years I had accumulated a thick stack of APL programs and utilities that I used in my work as a communications research psychologist. These had been my everyday tools, but would not run on my PC. I was hungry to once again sink my teeth into APL.

So when PC asked me to take a look at an APL package produced by STSC, I immediately canceled a long-awaited cross-country skiing trip and closeted myself with the manuals. Skiing could wait; APL was my first programming language and an exceedingly powerful one, and I was eager to see how well it was implemented.

THE BOOKS
are designed for
beginners and are
exceptionally well put
together.

STSC, the company that produced the APL*PLUS/PC package, is the leading supplier of APL services nationwide. It was founded in 1969 as a computer time-sharing organization specializing in corporate financial management systems and other APL development services.

Headquartered in Rockville, Maryland, STSC has offices in 21 cities in the United States and Europe, and provides educational and consulting services as well as user support. The company had previously released a stand-alone APL package for the TRS-80, but its implementation for the IBM PC is far more advanced. As a result, the APL*PLUS/PC system offers much faster program execution time and runs much larger programs than previous versions.

The package includes a disk, several books and manuals, and a substitute ROM chip containing the more than 50 special APL symbols at the heart of the language. On one side of the disk are the required APL systems files; turn it over and you find several demos and utilities. STSC provides the user with two excellent books on the APL language, and a fat 3-

Figure 2: Comparison of APL*PLUS/PC and BASIC in computing the mean, standard deviation, maximum and minimum of a series of numbers in floating point.

Number of Seconds to Execute			
Number of Numbers	BASIC	APL	Difference Factor
100	5.3	0.7	8
500	17.5	2.9	6
1,000	32.9	5.7	6
5,000	156.6	29.1	5

Figure 3: Comparison of APL*PLUS/PC and BASIC in sorting a series of numbers.

Number of Seconds to Execute			
Number of Numbers	BASIC	APL	Difference Factor
100	20	0.3	67
200	44	0.6	73
500	120	1.2	100

Figure 4: APL Programs Used In The informal time tests above.

```
▽ LOOPCOUNT N
[1] A COUNTS TO N USING LOOP
[2] C←0
[3] START:→(N=C←C+1)/END
[4] →START
[5] END: 'DONE'
[6] ▽

▽ RHOCOUNT N
[1] pNpl A CREATES VECTOR OF N 1S THEN GIVES VECTOR SIZE
[2] ▽

▽ SUMTO N
[1] SUM OF NUMBERS FROM 1 TO N
[2] +/N
[3] ▽
```

THE EXISTING APL
community may swarm to APL*PLUS/PC
like bees to honey.

ring binder explaining its particular implementation. The books are designed for beginners and are exceptionally well put together. The manual is aimed at experienced APLers, and is not up to the standards of the rest of the package.

Under The Hood

This weakness becomes apparent as soon as you begin to install the new ROM. Installation instructions are buried at the back of an appendix. The manual tells you to look for the character chip you are to replace by examining the tiny numbers stamped on otherwise identical chips under the hood. In my PC, many of the ROM chips had blurred or missing ID

STSC IS
*the leading supplier of
APL services
nationwide.*

numbers, and I had to start pulling out boards to examine them all. There were no diagrams, and the directions were fairly confusing. After half an hour of puzzling over which chip to remove, I called the Rockville office, where I received prompt, gracious help. It turned out that the chip I was looking for was on the monochrome adaptor board and not the mother board itself. They admitted that others had complained about these instructions.

However, once I found the location of the ROM, installation was a snap. Perhaps I should have known better, but why not make it easy for people and be clear about such things in the first place?

I was concerned that the substitute character set in ROM would turn my PC into an APL-only machine. Would I be able to use EasyWriter to write this article? Would this prevent my two sons, David and Daniel, from playing Decathlon? Would I still be able to use BASIC? I booted up APL with bated breath.

Everything worked perfectly. The special APL characters replace only the most obscure and least needed characters in the IBM set. You lose the four playing card suits and other low-demand charac-

ters such as the lower case y with an umlaut above it. You still have 256 characters, including upper- and lowercase alphabets; virtually all of the French, German, and Spanish special characters; and all the punctuation you could want. The 96 ASCII characters and the 48 line drawing symbols are not altered.

The package will work on any printer normally supported by the PC. However, you will need Grafrax or its equivalent on your Epson/IBM printer to print out the APL symbols. And there is no problem with ordinary text; you can print anything you could before you started using APL.

APL*PLUS/PC runs under DOS, and both systems fully reside in memory. As this takes up 90K, however, the minimum system requirement is 128K. The system can load any DOS files from within APL, which lets you pull in VisiCalc tables, for example, then manipulate the data in APL and send the revised tables back to the disk or to a remote location in VisiCalc form. And it allows the PC to function either as a dumb or a very smart terminal. It even enables you to download data onto your screen from a remote source via modem and then pull this information off the screen into memory.

Thinking Mathematically

Rather than just a computing language, APL, which stands for A Program-

THE
APL*PLUS/PC system
*offers much faster
program execution time
and runs much larger
programs than previous
versions.*

ming Language, is an algorithmically-oriented mathematical notation system that both the user and computer can understand. When you learn BASIC or Pascal you learn a computer language. When you learn APL you learn a language that you and the computer share. And, not

trivially, you learn a language that helps you think mathematically.

Programming in APL makes use of a large number of special symbols. Some of these such as + or - or $\times$ or $\div$ are used the conventional way. However, most of the APL operators provide powerful functions not normally available. For instance, the symbols $\uparrow$ and $\downarrow$ are used to sort, $\lceil$ and $\lfloor$ to find the maximum and minimum, $+/$ and $+\backslash$ to do sums and cumulative sums, and $\boxdiv$ for matrix division to solve simultaneous equations given a set of variables on one side and coef-

DEVOTEES OF
APL claim that users
*can develop APL
programs four to ten
times faster in APL
than in FORTRAN or
BASIC.*

ficients on the other. Other symbols allow indexing, searching, and various relational operations.

These various "primitive" functions are extremely powerful tools that allow you to write programs using extremely concise code. Devotees of APL claim that users can develop APL programs four to ten times faster in APL than in FORTRAN or BASIC.

When I first began working with computers in 1953 for a series of research studies at NYU on teenage drug use, I wanted to include advanced multivariate analyses using six 75×75 matrices, and commissioned IBM to write a program that would do it. It took IBM a full year and cost \$1,500 (remember, this was 30 years ago). The same program in APL can be written in one line:

```
▽R←X CORR Y  
[1] R←AVG(STD X)×STD Y  
▽
```

This program is set up as a user-defined function that takes both a left and a right argument, and yields an explicit result that can be displayed, stored, or incorporated into other functions. It uses

the functions AVG and STD, which must be in your active workspace. APL is ideally suited for such modular programming.

Short And Sweet

Because APL allows you to be concise, APL programs are generally very short. There was a story going around the Yorktown Heights Research Center years ago that the original programming for all of BASIC was written in one day using APL.

APL*PLUS/PC gives you an active workspace of about 41K with a 128K system. If you have a 512K system, the workspace increases to 420K. Stored workspaces, each with its own user-supplied name, are available from library disks.

While STSC recommends using an 8087 co-processor, my impression is that this would be necessary for unusually large number-crunching jobs only. Even without the co-processor, APL is impressively fast.

Before I received a promised set of benchmark results from STSC, I performed several informal tests of my own (see Figure 1). The first looked at count-

ONCE I
found the location of
the ROM, installation
was a snap.

ing, which can be done two ways in APL—by a loop, as is also done in BASIC, or by creating a “vector” of a certain number of 1s and then using the shape function ρ , which returns the size of an array. Looping from 1 to 5,000 took 92.7 seconds, while the vector method took a mere 0.3 seconds. Keep in mind that looping is seldom needed in APL.

In a second informal test, I calculated the cumulative sum of all the integers from 1 to 5,000. Here I used the expression $+/1N$, which creates a vector of progressive whole numbers from one to N and then sums the numbers in the vector. APL*PLUS/PC sprinted through 5,000 integers in 3.7 seconds.

You can perform many operations in

APL without writing a full-blown program by using its “immediate execution mode.” APL*PLUS/PC turns your PC into what is probably the world’s most powerful stand-alone desk calculator.

WHEN YOU
learn APL you learn a
language that you and
the computer share.

APL-Pie Order

Rich Paulson at STSC sent me the results of several benchmark tests he and his colleagues performed to compare APL to BASIC on a PC without an 8087. He measured trapped elapsed time without printing. Figure 2 shows the results for developing the mean, standard deviation, maximum and minimum of a series of N floating point numbers. BASIC chugged through in 156.6 seconds, while APL*PLUS/PC took 29.1.

Figure 3 compares APL and BASIC in sorting three series of numbers. STSC used the Quick Sort from Dwyer and Critchfield’s BASIC and the Personal Computer, which is reputed to be the fastest sorting routine available in BASIC. Figure 4 shows the APL programs I used to sort the numbers. It took BASIC 120 seconds to sort 500 numbers. APL*PLUS/PC simply blew BASIC away by arranging all 500 in perfect order in only 1.2 sec-

FORMER BASIC
users often grow a bit
rhapsodic when
describing some of
APL’s most useful
features.

onds. Here again APL doesn’t need a program. It can do the sort in two short steps in immediate execution mode. The raw

numbers are entered as a vector and assigned the name A. Then the upgrade primitive is used by entering $A[\uparrow A]$, and almost before you take your finger off the Enter key, the entire sort is done.

Former BASIC users often grow a bit rhapsodic when describing some of APL’s most useful features, and with good reason. It is crammed to the gunwales with niceties that make BASIC look like a toy.

Arrays in APL do not have to be declared, as no storage need be allocated in advance. Arrays of two or more dimensions are defined when first used in a program, and can be easily reshaped at any time under program control.

There are dozens more primitive operators in APL than BASIC. These operate on entire arrays simultaneously without requiring time-consuming loops. This ap-

EXPERIENCED
APLers will be happy
to hear that
APL*PLUS/PC is a very
complete
implementation of the
language.

plies to multidimensional matrices as well as scalars and vectors.

APL automatically stores values from -32,678 to 32,767 as whole numbers. If one element of an array is not an integer, the entire array is converted to floating point numbers. Results are normally printed to 10 significant digits, but can be displayed in up to 16.

Lines of code can be given mnemonic labels that remain constant during renumbering. This is very useful in GOTO situations and in the few cases when you need to loop. Line numbering is simple and efficient; to add a line between [1] and [2] you simply use [1.1]. To insert a line between [1] and [1.1] you use [1.01] and so on. These are automatically renumbered into integers when the program is completed. In program listings, comments are labeled and automatically

offset one space to the left for easier reading. Variable names can be up to 77 characters long. The names for character strings and numerical variables do not have to be distinguished.

IT **ALLOWS** *the PC to function either as a dumb or a very smart terminal.*

Errors Made Easy

When a program bombs out, APL not only gives the error message and prints the program line, but points to the exact location where the problem was encountered. BASIC users often have to puzzle this out for themselves.

Users can be assigned different password numbers for security or accounting purposes. There are many other security features, such as locked functions and the automatic expunging or modification of workspaces if someone other than the specified user loads it.

Experienced APLers will be happy to hear that APL*PLUS/PC is a very complete implementation of the language. All the APL operators are included, and there is a number of significant STSC enhancements. Among these are its shared file system, the availability of over 100 systems functions, a powerful formatting and exceptions handling capacity, and the ability to create and use onscreen windows.

T **HE MANUAL** *could stand improvement, especially for beginners.*

The package does have several shortcomings. As mentioned above, the manual could stand improvement, especially for beginners. In addition to the problem I

had installing the chip, I couldn't find any information on how to assign user-defined entry sequences to the 30 available function keys. I eventually learned that this is covered on the reverse side of the disk, but there is no explanation in the manual of what is on the disk.

The manual is also short on indexing. A large section on the more than 100 systems functions has no separate table of contents. And there is almost nothing in the manual on APL's powerful format function.

Alien Strokes

STSC does provide a sturdy laminated guide to the keyboard's new configuration of Shift, Alt, and Ctrl key functions. But perhaps it should make available a set of covers to fit over the actual keys. For beginners, using these alien keystrokes can be an extremely slow process. STSC should also issue a handy reference guide listing the operation and function of these keys.

STSC seems committed to publishing the most comprehensive APL package on the market. Enhancements are issued regularly, and are offered to users at a nominal charge. Future releases are expected to include a keyword version based on English words rather than APL symbols, for users who do not want to install the character chip or deal with the alphabet of symbols. STSC is also working on a Help command that will provide on-line programming assistance.

STSC has a stated policy that there will be nothing that BASIC can do that APL*PLUS/PC cannot do equally well. Music and color graphics functions and routines are scheduled to become a part of the system, and the entire kit will soon be boxed and sized to resemble IBM's manuals.

Under consideration for later release are a financial planning system with spreadsheet capability, a graphics and plotting package that will also include a screen management system, and a data file management interface system for users who know nothing about APL. An inexpensive statistical program may soon be issued. Software writers will be glad to hear that STSC will soon put a run-time version of APL on the market for application development. While applications with this system will run under APL, end users will not have access to the language

T **HE** *"primitive" functions are very powerful tools that allow you to write programs using extremely concise code.*

itself.

The existing APL community may just swarm to APL*PLUS/PC like bees to honey. The system should also appeal to scientists, engineers, actuaries, mathematicians, statisticians, finance experts, and anyone else who wants an extremely fast and powerful computational capability at his fingertips.

But APL is not simply an expensive calculator. Many users feel it is the most advanced general purpose computer programming language available today. Programming in APL is easy to learn; with enough guidance, most APL beginners can function productively after one session with it. And the educational materials provided by STSC are excellent.

My own personal reaction was one of awe and delight. Even without an 8087 co-processor, APL*PLUS/PC transformed my PC into a lightning-fast powerhouse. STSC is to be congratulated on

M **ANY USERS** *feel APL is the most advanced general purpose computer programming language available today.*

its sterling implementation; APL*PLUS/PC is a major accomplishment that may influence the future of microcomputing.

/PC

stsc

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000